# CS 255 Business Requirements Document Template – Ryan Stork

Complete this template by replacing the bracketed text with the relevant information.

This template lays out all the different sections that you need to complete for Project One. Each section has guiding questions to prompt your thinking. These questions are meant to guide your initial responses to each area. You are encouraged to go beyond these questions using what you have learned in your readings. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead, the goal is to complete each section based on your client’s needs.

**Tip:** You should respond in a bulleted list for each section. This will make your thoughts easier to reference when you move into the design phase for Project Two. One starter bullet has been provided for you in each section, but you will need to add more.

## System Components and Design

### Purpose

*What is the purpose of this project? Who is the client and what do they want their system to be able to do?*

* Our client, DriverPass, wants to create an application to help drivers prepare for a DMV Driver’s License Test.
* In our interview, we were asked to design a web-based app for DriverPass to help accomplish their goals.

### System Background

*What does DriverPass want the system to do? What is the problem they want to fix? What are the different components needed for this system?*

* DriverPass wants users to help people pass a driver’s license test.
* We will create an online application to set up driving lessons.
* Also, to have online classes and practice tests.

### Objectives and Goals

*What should this system be able to do when it is completed? What measurable tasks need to be included in the system design to achieve this?*

* Upon completion, a user should be able to create an account or log into an existing account
* The user should be able to change their password, if needed.
* The user should be able to purchase a package
* The user can arrange a driving lesson online, by phone or in-person as well as edit or cancel those lessons
* From the interview, we need to have a database and calendar for driving lessons.
* We need to have different levels of security access based on role with the company or user.
* We need to have different packages available for a user to choose. We also need the ability to disable a package if it is determined that it is not needed.
* We need a tracking system to know who added, changed, or deleted a driving lesson
* We need to pair a driver to a user.
* We need the system to be cloud-based so that a user can take a lesson anywhere, anytime.
* A second reason for a cloud-based system is so that DriverPass doesn’t have to worry about maintenance or information security, it is run by the cloud company.
* The user needs the ability to download training documents to prepare for online classes.
* When the DMV updates the driving test, DriverPass is notified so they can make changes
* Drivers will need the ability to add comments after a driving lesson.
* Driving lessons are two hours, we must make sure that you cannot overbook a driver.

## Requirements

### Nonfunctional Requirements

*In this section, you will detail the different nonfunctional requirements for the DriverPass system. You will need to think about the different things that the system needs to function properly.*

#### Performance Requirements

*What environments (web-based, application, etc.) does this system need to run in? How fast should the system run? How often should the system be updated?*

* The Web environments that are needed to run are different web browsers, like Chrome, FireFox, and Safari.
* Since the application is also for mobile devices it needs to run on Android and iOS.
* Since the system is cloud-based, it might be cheaper to start up by renting cloud servers, like AWS. This would allow our speed to be the current speed of their servers and we wouldn’t have to worry about the initial price of purchasing servers.
* The system should be updated each time a user makes a change. If these changes affect other users, then they should be notified. This can be done with a notification symbol on their page to let them know that something was updated.

#### Platform Constraints

*What platforms (Windows, Unix, etc.) should the system run on? Does the back end require any tools, such as a database, to support this application?*

* AWS (Amazon Web Services) run on Linux. This frees the user from paying for Microsoft windows applications. Many Linux tools are free. Our devices should connect to the server from anywhere.
* For the system administrators, I would give extra access with database maintenance tools and spreadsheet tools for any maintenance to the application that they may need to make.

#### Accuracy and Precision

*How will you distinguish between different users?* *Is the input case-sensitive? When should the system inform the admin of a problem?*

* The different users are differentiated at the log in. Only the owner will be able to create system administrator, secretary, driver accounts. If an account is made by someone other than the owner, then it is automatically seen as a user.
* I would not have the usernames as case sensitive and possibly an email address for usernames, this will make sure that all users can be contacted. The passwords will be case sensitive.
* The system should inform the admin of a problem when multiple users are unable to access their accounts in a short span of time. This may show that there is a problem with the server rather than with the users.

#### Adaptability

*Can you make changes to the user (add/remove/modify) without changing code? How will the system adapt to platform updates? What type of access does the IT admin need?*

* A system administrator or owner will be able to add/ remove/ or modify basic users. Only the owner will be able to modify other accounts. This will ensure that only the owner can make changes to the team members. There would be no need to change code because these functions would be built in.
* Before AWS gets updated, they send out notices to the users of that platform of the changes. This allows our program to update before the system update goes live.
* The IT team would need remote access to the servers to make any updates as necessary.

#### Security

*What is required for the user to log in? How can you secure the connection or the data exchange between the client and the server? What should happen to the account if there is a “brute force” hacking attempt? What happens if the user forgets their password?*

* For the user to log in you would be required a username and password. The username would be an email address.
* The password would be sent to the server as a hash of the user input. In our database, we would only have a copy of the hash, not the actual password. Then we would check if the passwords match.
* If 3 incorrect inputs are entered, then they would have to change their password after they are asked a security question. If the security question is answered correctly, then the user can put in for a new password. The same would happen if the user forgets their password.
* If the user is an employee, they would also have to log in with two-factor authentication. This would be done with an authenticator like Google Authenticator. This would prevent unauthorized entry on employee accounts.

### Functional Requirements

*Using the information from the scenario, think about the different functions the system needs to provide. Each of your bullets should start with “The system shall . . .” For example, one functional requirement might be, “The system shall validate user credentials when logging in.”*

* The system shall allow be able to change their password, if needed.
* The system shall allow a user should be able to purchase a package
* The system shall verify different levels of security access based on role with the company or user.
* The system shall allow different packages available for a user to choose.
* The system shall have ability to disable a package if it is determined that it is not needed.
* The system shall have a tracking system to know who added, changed, or deleted a driving lesson
* The system shall pair a driver to a user.
* The system shall allow the ability to download training documents to prepare for online classes.
* The system shall make sure that you cannot overbook a driver.

### User Interface

*What are the needs of the interface? Who are the different users for this interface? What will each user need to be able to do through the interface? How will the user interact with the interface (mobile, browser, etc.)?*

* The interface needs to differentiate the difference between the different users.
* The different levels of users need to have different access to the system.
* A standard user would be able to sign up for a package, download class documents, and set up driving lessons. This could be done via web browser or mobile device.
* A driver account would need to make comments on user accounts. This would be done via web browser.
* Secretary or team member accounts would need to be able to adjust standard user accounts if there is an issue. This would be done via a web browser.
* System Admins should have access to make corrections on any adjustments that were made incorrectly. This would be done via web browser
* Owner account should be able to add/ remove/ delete any account or make any changes necessary. This would be done via web browser.

### Assumptions

*What things were not specifically addressed in your design above? What assumptions are you making in your design about the users or the technology they have?*

* One assumption is that the user has access to the internet
* A second assumption is that the user lives in a city where we are offering services
* A third assumption is that the user is old enough to get a driver’s license.

### Limitations

*Any system you build will naturally have limitations. What limitations do you see in your system design? What limitations do you have as far as resources, time, budget, or technology?*

* There would be limitations with initial packages. We do not know what users will want. We should be able to add or delete packages as needed.
* More time should be given to develop the interface to make sure that it will work on multiple platforms simultaneously.
* We are limited on technology by the current systems that are available to us.

### Gantt Chart

*Please include a screenshot of the GANTT chart that you created with Lucidchart. Be sure to check that it meets the plan described by the characters in the interview.*

Timeline

Description automatically generated